

**VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra Informatiky**

**Srovnání kvality virtuálních strojů  
Assessment of Virtual Machines Performance**

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě dne 20. dubna 2011

.....  
Lenka Novotná

Ráda bych poděkovala vedoucímu bakalářské práce, Ing. Petru Olivkovi, za pomoc, věcné připomínky a veškerý čas, který mi věnoval.

## **Abstrakt**

Hlavním cílem této bakalářské práce je vysvětlit pojem virtualizace, virtuální stroj, a dále popsat a porovnat dostupné virtualizační prostředky, které se využívají ve světovém oboru informačních technologií.

Práce se především zabývá srovnáním výkonu virtualizačních strojů pro desktopové operační systémy MS Windows a Linux. Při testování jsem se zaměřila na tři hlavní faktory a to propustnost sítě, při které jsem použila aplikaci Iperf, ke změření výkonu diskových operací jsem využila program IOZone a pro test posledního faktoru, který je zaměřen na přidělování CPU procesů, jsem použila známé testovací aplikace Dhrystone a Whetstone. Všechna zmínovaná měření okruhů byla provedena na třech virtualizačních platformách, kterými jsou VirtualBox OSE, VMware Player a KVM s QEMU.

### **Klíčová slova:**

Virtualizace, virtuální stroj, VirtualBox, VMware, KVM, QEMU, plná virtualizace, paravirtualizace, částečná virtualizace, hardwarově asistovaná virtualizace, virtualizace na úrovni operačního systému, měření výkonu CPU, měření propustnosti sítě, měření diskových operací, Dhrystone, Whetstone, Iperf, IOZone.

## **Abstract**

The main goal of this thesis is explain the term of Virtualization and Virtual Machine, describe and compare available virtualization resources, which we can use in worldwide field of information technology.

The purpose of this thesis is assessment virtual machine performance for desktop operation system like MS Windows and Linux. During testing I focused for three main factors – network throughput, disk operations and CPU process. Network throughput was measured by Iperf application. For test small and large disk operations I used IOZone benchmark tool and the last test measuring CPU process with benchmark application Dhrystone and Whetstone. This measurements were executed on three platform of virtualization – VirtualBox OSE, VMware Player and KVM with QEMU.

### **Keywords:**

Virtualization, virtual machine, VirtualBox, VMware, KVM, QEMU, full virtualization, paravirtualization, hardware-assisted virtualization, operating system-level virtualization, partial virtualization, CPU benchmark, network testing, file system benchmark, Dhrystone, Whetstone, Iperf, IOZone.

# Seznam použitých symbolů a zkratek

**CPU** Central Processing Unit

**BIOS** Basic Input-Output System

**GUI** Graphical User Interface

**OVF** Open Virtualization Format

**TCP** Transmission Control Protocol

**UDP** User Datagram Protocol

**P2P** Peer-to-peer

**IP** Internet Protocol

**PUEL** Personal Use and Evaluation Licence

**GPL** General Public Licence

**OSE** Open Source Edition

**VDI** Virtual Disk Image

**VMDK** Virtual Machine Disk Format

**NAT** Network Address Translation

**UTP** Unshielded Twisted Pair

**NAT** Network Address Translation

**DHCP** Dynamic Host Configuration Protocol

**DNS** Domain Name System

# Obsah

1. Úvod.....	1
2. Virtualizace .....	2
2.1 Virtuální stroj.....	2
2.2 Důvody nasazení virtualizace.....	4
2.3 Historie.....	5
2.4 Druhy virtualizace.....	5
2.4.1 Plná virtualizace (Full virtualization).....	6
2.4.2 Paravirtualizace (Paravirtualization).....	7
2.4.3 Hardwarově asistovaná virtualizace (Hardware-assited virtualization) 8	
2.4.4 Virtualizace na úrovni operačního systému (Operating system-level virtualization).....	9
2.4.5 Částečná virtualizace (Partial Virtualization).....	9
3. Praktická část.....	11
3.1 Testované virtualizace platformy.....	11
3.1.1 VirtualBox.....	11
3.1.2 VMware Player.....	13
3.1.3 KVM a QEMU.....	13
3.2 Příprava testů.....	14
3.2.1 Přidělování CPU procesům.....	14
3.2.2 Diskové operace.....	15
3.2.3 Propustnost sítě.....	18
3.3 Měření a vyhodnocení.....	18
3.3.1 Měření přidělování CPU procesům.....	18
3.3.2 Propustnost sítě měřená pomocí Iperf aplikace.....	22
3.3.3 Diskové operace měřené pomocí IOZone.....	26
4. Závěr.....	32
5. Použité zdroje.....	33

## Seznam obrázků a grafů

Obrázek 1: Vrstvy nevirtualizovaného počítače.....	9
Obrázek 2: Vrstvy virtualizovaného počítače s hypervisorem pro serverové operační systémy..	9
Obrázek 3: Vrstvy virtualizovaného počítače s hypervisorem pro desktopové operační systémy	9
Obrázek 4: Privilegovaný režim procesoru.....	10
Graf 1: Whetstone Linux.....	25
Graf 2: Whetstone Windows.....	26
Graf 3: Dhrystone Linux.....	27
Graf 4: Dhrystone Windows.....	28
Graf 5: Iperf Linux NAT.....	30
Graf 6: Iperf Windows NAT.....	31
Graf 7: Iperf Linux Bridge.....	31
Graf 8: Iperf Windows Bridge.....	32
Graf 9: Linux Write 128km.....	33
Graf 10: Windows Write 128k.....	34
Graf 11: Linux Read 128k.....	34
Graf 12: Windows Read 128k.....	35
Graf 13: Linux Write 4G.....	35
Graf 14: Windows 4G.....	36
Graf 15: Linux Read 4G.....	36
Graf 16: Windows Read 4G.....	37

# 1. Úvod

Virtualizace je velmi oblíbenou technologií nejen ve velkých firmách, ale je také využívána v malých rozrůstajících se podnicích a to především kvůli nízké pořizovací ceně kompletního řešení virtualizovaného prostředí.

Tato bakalářská práce se dělí na dvě části. Hlavním cílem první části je především vysvětlit důležité pojmy, které souvisí s pojmem virtualizace a podat jasný přehled o využití této techniky, dále porovnat virtualizované a nevirtualizované stroje a objasnit jakým způsobem je využíváno ochranných kruhů procesoru s používáním virtualizace. V jednotlivých podkapitolách se detailněji věnuji popisování aktuálních virtualizačních technik a vysvětlení jakým způsobem jsou ovlivněny virtualizované obrazy. Dále se zabývám popisem nejznámějších důvodů jejího používání, srovnání výhod a nevýhod. Ke každé technice uvádím nejznámější produkty. Práce je především věnována virtualizačním strojům pro desktopové aplikace.

V druhé části se zabývám porovnáním třech virtuálních strojů testovanými na dvou operačních systémech. Testovanými produkty, které jsou volně dostupně pro uživatelské používání jsou VirtualBox, VMware a KVM s QEMU. Při testování jsem se zaměřila na tři různé testy a to práci procesoru, propustnosti sítě a měření diskových operací. Vybrané nejvhodnější testovací aplikace jsem rozdělila do podkapitol a uvedla, jakým způsobem pracují. Testovací aplikace jsou kompatibilní se všemi stroji. Po změření těchto třech kritérií jsem výsledky přehledně zpracovala a uvedla je v grafech. Na závěr této práce jsem shrnula naměřené výsledky.



## 2. Virtualizace

S pojmem virtualizace se v dnešní době setkáme především v oblasti informačních technologií. Je to vlastně způsob, kdy využíváme fyzického stroje (ať už pouze hardwarové komponenty jako je procesor, počítačová paměť, zapojení pevných disků, či dokonce kompletní počítač) nebo pouze softwarového prostředí k tomu, abychom je mohli pomocí určitého druhu virtualizace používat jiným způsobem, než jakým by fyzicky měli pracovat.

### 2.1 Virtuální stroj

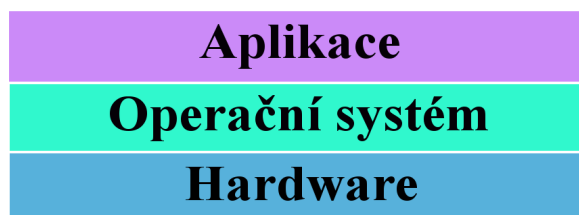
Než začneme s popisováním jednotlivých druhů virtualizací, bylo by vhodné si vysvětlit základní pojmy zbývající se touto problematikou jako jsou virtuální stroj, virtualizace platformy, hypervisor a virtuální obraz.

Virtualizace platformy se provádí na určitých hardwarových prostředcích pomocí řídicího programu, často také nazývaného hypervisor. Tato softwarová vrstva, která poskytuje virtualizaci, pracuje mezi fyzickým a virtuálním strojem a vytváří simulované prostředí pro hostující stanice, které se také nazývají virtuálními obrazy. Na takový virtuální stroj můžeme nainstalovat několik hostujících stanic, které mohou pracovat a vykonávat programy jako skutečný stroj a přesto nemají žádný přístup k fyzickému hardwaru.

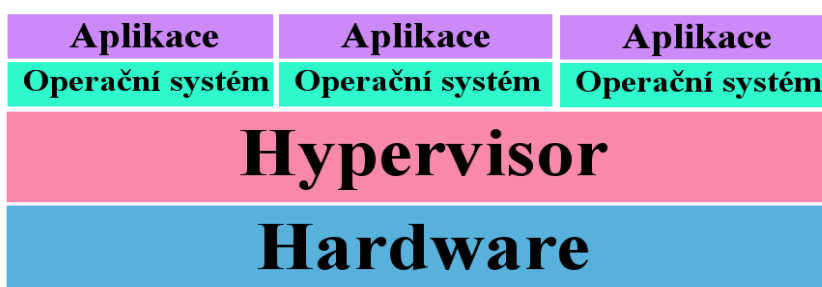
Hostující stanice ve většině případů neomezují uživatelský běh aplikací ani celkový výkon kompletního operačního systému a jsou vykonány přesně tak, jako kdyby pracovaly na aktuálním fyzickém hardwaru.

### Porovnání počítačových vrstev

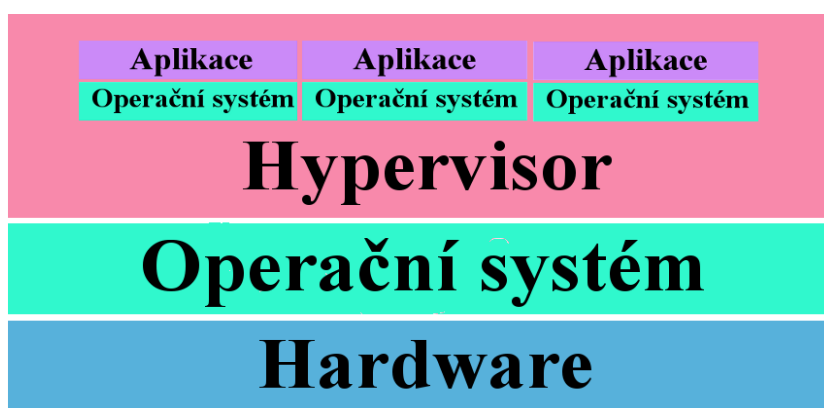
Na obrázku č. 1 vidíme klasické uspořádání vrstev pro řízení dnešních osobních počítačů. Při používání virtualizační techniky s hypervisorem se toto uspořádání změní. Existují dva druhy použití hypervisoru. První typ, který je zobrazen na obrázku 2 se používá především pro serverové operační systémy. Hypervisor běží přímo na hardwaru počítače a řídí tak hardware a hostované operační systémy. Obrázek 3 nám zobrazuje klasickou architekturu, kdy využíváme desktopové operační systémy. V tomto případě se hypervisor chová jako aplikace, která běží uvnitř prostředí operačního systému.



Obrázek 1: Vrstvy nevirtualizovaného počítače



Obrázek 2: Vrstvy virtualizovaného počítače s hypervisorem pro serverové operační systémy



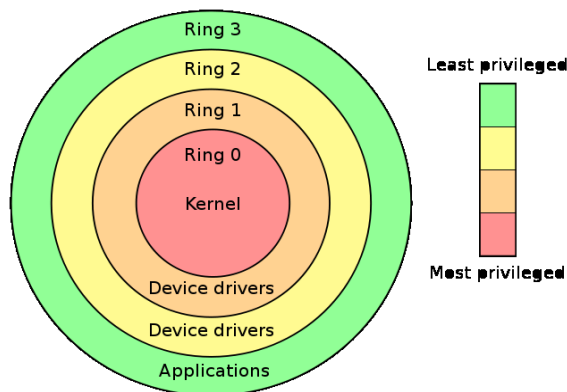
Obrázek 3: Vrstvy virtualizovaného počítače s hypervisorem pro desktopové operační systémy

## Práce procesoru a hierarchie privilegovaných kruhů

Procesorový běh zahrnuje práci v privilegovaném režimu a zavádí takzvané ochranné kruhy (ring). Tento ochranný mechanismus slouží pro řízení správné funkcionality a tím se vyvarovává nechtěnému používání a chrání tak data a zaručuje bezpečný běh počítače.

V závislosti na architektuře počítačového systému je poskytováno několik stupňovaných vrstev ochranných kruhů, což můžeme vidět na obrázku 4, kde jsou kruhy seřazeny od nejprivilegovanějšího režimu k nejméně privilegovanému. Většina operačních systémů pracuje se čtyřmi kruhy, kde kruh 0 (Ring 0) pracuje v nejvíce privilegované vrstvě a přímo působí na fyzický hardware, jako je procesor a paměť, a tedy vykonává privilegované

instrukce. V tomto kruhu je umístěno jádro operačního systému, které se nazývá supervisor mode. Kruhy 1 a 2 (Ring 1 a Ring 2) jsou většinou nevyužité (platí pro architekturu x86). Třetí kruh (Ring 3) se nazývá user mode, tedy uživatelský mód a řídí práci uživatelských programů, které nemají plný přístup k hardwaru.



Obrázek 4: Privilegovaný režim procesoru

Zdroj: [http://en.wikipedia.org/wiki/File:Priv\\_rings.svg](http://en.wikipedia.org/wiki/File:Priv_rings.svg)

Při virtualizaci tak dochází ke změně využívání těchto vrstev. Aby se virtualizované operační systémy vzájemně neovlivňovaly, jejich jádro se přeneso do kruhu 1 (Ring 1) a do kruhu 0 (Ring 0) se zavede hypervisor. Hlavním úkolem hypervisoru je práce a obsluha procesoru a jeho privilegovaných instrukcí, práce s přidělováním paměti a se vstupními a výstupními zařízeními jako jsou pevné disky, síť a nebo také může poskytovat rozhraní pro volání paravirtualizace.

## 2.2 Důvody nasazení virtualizace

Nejznámějším případem použití virtualizace je spojení několika menších serverů. Tyto malé fyzické stroje jsou nahrazeny jedním výkonnějším serverem, tím pádem ušetříme za nákup nákladných hardwarových zdrojů jako je například procesor. Tímto způsobem se efektivně spouští odlišné operační systémy běžící uvnitř virtuálního stroje. Jeden velký server tak tedy může hostovat spousty jiných, souběžně pracujících, hostovaných strojů. Virtuální stanice mohou být řízeny a kontrolovány ze vzdáleného fyzického stroje.

Dalším důvodem využití této technologie může být její prospěch při vývoji počítačového jádra (kernel), nebo při testování funkčnosti implementovaných produktů na různých platformách ve firmě zabývající se vývojem softwaru.. Samozřejmě je ho možné také využít pro vyučování a názornou ukázkou různých počítačových operačních systémů.

Hlavní výhodou využívání virtualizačních nástrojů je přenášení jednotlivých hostových stanic. Obrazy virtuálních strojů mohou být přenášeny z jednoho fyzického stroje do jiného. Kompletně nainstalované a nakonfigurované stanice se dají klonovat, kopírovat a nebo

exportovat do OVF formátu, který je použitelný pro většinu dostupných produktů (VirtualBox, Red Hat Enterprise Virtualization, VMware ESX, Workstation Player). To nám ušetří spoustu práce s novou instalací.

Další velkou výhodou je práce na operačním systému ve virtuálním stroji bez zásahu do fyzického stroje, nebo při spuštění aplikací, které nejsou podporovány pro fyzický stroj. Jestliže nastane chyba ve virtuálním stroji a selže operační systém, fyzický stroj zůstane v pořádku a nedotčený.

Používání virtualizačních technik sebou přináší také velkou výhodu v ceně. Není potřeba nakupovat další hardwarové komponenty pro instalaci operačních systémů a tím se sníží i náklady na údržbu a elektřinu, proto říkáme, že virtualizace je šetrná k přírodě – GREEN IT.

Mezi nejzřejmější nevýhody bych zařadila snížení výkonu počítačových stanic, selhání fyzického stroje, ať už hardwarově nebo softwarově (řešením je pravidelné zálohování) a problém s identickou IP adresou a názvem počítače při kopírování virtuálních strojů (řešením je například program od Microsoftu, který slouží pro přípravu operačního systému k pozdějšímu kopírování zvaný Sysprep).

## 2.3 Historie

Počátky virtualizace sahají do roku 1960 a pojem virtuální stroj, dříve nazývaný jako pseudo stroj, se poprvé objevil v pokusném systému IBM M44/44X. Později se tato technologie uplatnila pod názvem virtualizace platformy nebo serveru.

## 2.4 Druhy virtualizace

V této podkapitole jsou popsány dostupné virtualizační techniky. Jednotlivé druhy těchto technik jsou seřazené podle způsobu jejich používání. Tyto druhy jsou postupně charakterizované jakým způsobem využívají virtualizované prostředí a jejich práce s hypervisorem, dále jsou popsány hlavní výhody a nevýhody a uvedeny jejich dostupná řešení.

**Techniky virtualizace se dělí na:**

- Plná virtualizace
- Paravirtualizace
- Hardwarově asistovaná virtualizace
- Virtualizace na úrovni operačního systému
- Částečná virtualizace

### 2.4.1 Plná virtualizace (Full virtualization)

Plná virtualizace využívá určitý druh prostředí pro virtuální stroj, který je schopný kompletně simulovat hardwarové zdroje. Používá tedy speciální druh hostitelského softwaru, kterému se říká hypervisor. Ten vzájemně spolupracuje přímo s procesorem fyzického stroje a diskovým prostorem. V takovém prostředí je jakýkoliv software schopný používat základní hardwarové zdroje, které běží ve virtuálním stroji a to konkrétně jakýkoliv každý operační systém. Každý host běží na svém vlastní operačním systému. Operační systémy nemusí být modifikované, aby se daly použít a jsou úplně nezávislé. Výjimkou jsou některé platformy, které dovolují pouze určitý druh modifikovaného softwaru, který běží uvnitř virtuálního stroje. Hypervisor uchovává každý virtuální obraz úplně nezávislý od ostatních, nic netušících strojů běžících na jednom fyzickém stroji. Nejčastěji se využívá jako platforma pro operační systémy virtuálních serverů.

Hypervisor pracuje tak, že řídí a střídá hardwarové zdroje fyzického stroje příslušné virtuální stanici. Hypervisor má vlastní procesorové požadavky. To znamená, že fyzický stroj musí rezervovat část procesorové paměti, aby jej mohl hypervisor použít. To má však vliv na celkový výkon stroje a může zpomalit běh aplikací.

Dalo by se říct, že hypervisor je softwarová aplikace, která rozděluje jednotlivé fyzické stroje do několika virtuálních obrazů a tím poskytuje sjednocení serverů, což může být velmi prospěšné. Existující aplikace a spustitelné soubory tak pracují nepozměněny. Hypervisor formuje vrstvu mezi operačním systémem běžícím uvnitř virtuálního stroje a hardwarem. Toto oddělení poskytuje hypervisoru možnost kontrolovat jak hostitelské operační systémy pracují uvnitř virtuálního stroje a jak využívají hardwarových zdrojů.

Hypervisor také poskytuje jednotný přehled o hardwaru. Pokud používáme virtuální technologie různých výrobců a zároveň jiný fyzický hardware, znamená to, že virtuální stroj může běžet na jakémkoliv podporovaném počítači. Můžeme tak říci, že vidíme hardware jako uložisko dostupných hardwarových zdrojů, které mohou být používány pro jakoukoliv službu. Lze také používat živou migraci virtuálních obrazů přes jednotlivé počítače. Tato schopnost může být využívána při hardwarovém selhání fyzického stroje. Virtuální obrazy se dají snadno vypínat, nebo ho vrstva hypervisoru může přemapovat. Virtuální obrazy je také jednoduché replikovat a to nám dovoluje přinést nový obraz kdykoliv podle potřeb.

#### **Výhody:**

Plná virtualizace patří mezi virtualizace s nejlepší izolací a zabezpečením pro virtuální stroje a má nejjednodušší přenositelnost mezi ostatními stroji.

#### **Nevýhody:**

Fyzický server musí rezervovat část procesorové síly a paměti, aby mohl hypervisor řádně fungovat. To může mít vliv nad celkovým výkonem ať už fyzických, nebo virtuálních strojů a ve výsledku mít za následek zpomalení aplikací.

**Některá dostupná řešení:**

VirtualBox, Virtual PC, VMware Workstation, VMware Server, VMware ESX Server, QEMU, Parallels Desktop a Win4BSD

**2.4.2 Paravirtualizace (Paravirtualization)**

Dovoluje běh více operačních systémů na jenom hardwaru ve stejný čas a tím je možné využít větší množství systémových zdrojů jako je procesor a paměť. Na rozdíl od plné virtualizace, kde je emulovaný celý systém (BIOS, disk, procesor, síťová karta, *atd.*) řídicí modul (hypervisor, nebo také zvaný virtual machine monitor) řídí operační systém, který byl modifikován, aby řádně pracoval ve virtuálním stroji.

Virtualizované operační systémy tak budou při spouštění dosahovat lepšího výkonu než při plné virtualizaci, při které se musí vše emulovat. Paravirtualizace je tedy velmi efektivní díky její ceně a úrovni zabezpečení. Aby hlavní operační systém běžel paravirtualizovaný, musí být modifikovaný, to bohužel znamená, že ne všechny operační systémy a jejich distribuce jsou dostupné pro toto řešení. Např. Linuxová distribuce Red Hat, nebo Windows Server nemůžou být jako hlavní host pro toto řešení. Jádro těchto hostujících operačních systémů je přesunuto do ochranného kruhu 1 (Ring 1), a proto nemusí být modifikovány.

Cílem modifikovaného rozhraní je snížit část výkonu hostu, který tráví čas vykonáváním obtížných operací, tak aby běžel ve virtuálním prostředí, které je srovnatelné s nevirtuálním prostředím, tedy aby prostředí bylo co nejvíce podobné fyzickému. Paravirtualizace využívá hypervisor, který dovoluje virtuálním strojům, aby hlavní host potvrdil žádost na zpracování úloh, které by jinak byly vykonány ve virtuální doméně, kde je horší prostředí.

**Výhody:**

- Vysoký výkon

**Nevýhody:**

- Nelze virtualizovat odlišný hardware.
- Kód hostitelského operačního systému musí být modifikovaný.

**Některá dostupná řešení:**

Citrix Xen, VMware Server

### **2.4.3 Hardwarově asistovaná virtualizace (Hardware-assited virtualization)**

Je to virtualizační technika, která je vestavěna do základního hardwaru. Nutnou podmínkou pro běh této techniky je, aby hardwarové komponenty byly schopné tuto techniku využívat. Pracuje na architektuře x86 a poskytuje využití dodatečných hardwarových schopností, zejména procesor. Tento druh virtualizace byl poprvé použit v roce 1972. Hardwarově asistovaná virtualizace se oproti jiným technikám liší tak, že monitor virtuálního stroje má plnou kontrolu nad hardwarem počítače.

Výsledkem je virtualizovaná hardwarová podpora, která usnadňuje vestavěnému hypervisoru pracovat s virtuálními stroji výkonněji. Problémem této techniky je schopnost ovládat všechny privilegované instrukce a proto se používá ve spojení s plnou virtualizací nebo paravirtualizací. Každý virtuální operační systém může tedy pracovat izolovaně.

Nazývá se také zrychlenou virtualizací.

#### **Výhody:**

- Poskytuje jiný způsob přístupu do operačního systému.
- Operační systémy s architekturou x86 jsou navrženy tak, aby měly přímý přístup k systémovým zdrojům. S pomocí hardwarově asistované virtualizace má operační systém přímý přístup k systémovým zdrojům bez použití emulace nebo modifikace operačního systému.

#### **Nevýhody:**

- Požaduje systémy s novým druhem architektury x86-64 .
- Serverové sjednocení nemusí být vždy efektivní.

#### **Implementace pro hardwarovou asistenci:**

VMware Workstation (pouze pro 64bit) Xen, Linux KVM a Microsoft Hyper-V.

#### **2.4.4 Virtualizace na úrovni operačního systému (Operating system-level virtualization)**

Virtualizace na úrovni operačního systému je serverová virtualizační metoda, při které jádro operačního systému dovoluje práci více izolovaným uživatelským aplikacím, namísto jedné. Tyto aplikace systém vnímá jako skutečné. Jádro poskytuje zdrojové řídicí funkce izolovaně, aby omezil vzájemné ovlivnění jednotlivých strojů.

Také má schopnost živé migrace a může být používána pro dynamické načítání nastavení virtualizovaného obrazu mezi uzly v clusteru. Standardní souborový systém je sdílený mezi disky, které automaticky vytváří své kopie, což nám umožňuje jednoduché zálohování.

Program ve virtuálním disku používá klasické systémové volání operačního systému a nepotřebuje emulaci nebo prostředníka mezi virtuálním strojem jako v případě plné virtualizace, nebo paravirtualizace a také nepotřebuje hardwarovou asistenci.

Kromě výkonu je také velkou výhodou, že takový systém lze snadně spravovat a dynamicky měnit jeho nastavení. Není flexibilní jako ostatní virtualizace, protože nemůže hostovat rozdílné operační systémy.

##### **Výhody:**

- Velké množství prostředí bez ovlivnění operačních systémů
- Jednoduchá záloha.

##### **Nevýhody:**

- Nemůže hostovat operační systém s rozdílným jádrem.

##### **Implementace:**

FreeVPS, chroot, Linux-VServer, Container/Zone, Parallels Virtuozzo Containers

#### **2.4.5 Částečná virtualizace (Partial Virtualization)**

Částečná virtualizace simuluje některé ze základních hardwaru a k tomu používá virtualizaci adresního prostoru. Každý virtuální stroj se skládá z nezávislého adresního prostoru. Na virtuálním stroji tak může běžet většina nemodifikovaného softwaru, ale nelze spouštět úplné operační systémy. Některé operační systémy jako Windows a Linux mohou využívat tento druh virtualizace.

Podporuje sdílení zdrojů a izolaci procesů, ale nedovoluje oddělit hostované operační systémy.

Částečná virtualizace je předchůdcem plné virtualizace.



**Výhody:**

- Jednodušší implementace než při plné virtualizaci.
- Schopnost podpory důležitých aplikací.
- Vyznačuje se jako nejlepší způsob pro sdílení počítačových zdrojů více uživatelů.
- Vhodný pro DNS a hostující služby.

**Nevýhody:**

- Požaduje zpětnou kompatibilitu nebo přenositelnost. Pokud není hardware simulovaný, vystavujeme se riziku selhání softwaru.
- Je těžké předvídat některým aplikacím jejich vlastnosti.
- Operační systémy se nemohou spouštět ve virtuálním stroji, pouze aplikace .

## 3. Praktická část

V praktické části této bakalářské práce srovnávám výkon a kvalitu virtuálních strojů. Cílem této části je změření a následné porovnání virtuálních strojů s naměřenými výsledky fyzického stroje.

Příprava testů je důležitou součástí praktické části. Testy se řádně předem připraví, to znamená, že se vybere a vyzkouší potřebný software pro měření. Při měření se zaměřuji na tři hlavní způsoby testování a to přidělování CPU procesům, kdy zjišťuji výkon procesoru, následně testuji diskové operace a práce s diskem, především čtení a zápis pro malé a velké soubory. Poslední test se věnuje propustnosti sítě, což mi umožnilo porovnat výsledky virtuálního stroje oproti fyzickému.

Před tím, než se spustí testování virtuálních strojů je potřeba otestovat i fyzické stroje. V mém případě Linux s distribucí Ubuntu 10.04 a Windows 7 od Microsoft Windows. Test proběhl na každém fyzickém stroji s produkty Oracle VM VirtualBox a VmWare Player a pod Linuxem KVM (Kernel-based Virtual Machine). Ve výsledku tak vzniklo 10 různých kombinací testů.

### 3.1 Testované virtualizace platformy

Používanými aplikacemi pro testování byly volně dostupné virtualizační nástroje pro desktopové osobní počítače a to VirtualBox, VMware Player a KVM.

Cílem této podkapitoly je popsat tři již zmíněné techniky, podat jasný přehled jakým způsobem je ovlivněno prostředí fyzického stroje, jaké jsou možnosti využití a nastavení virtualizačního hardwaru a dále popsat jaké jsou jejich hlavní zajímavosti a výhody.

#### 3.1.1 VirtualBox

Oracle VM VirtualBox je virtualizace pro architekturu x86 původně vytvořená německou softwarovou společností Innotek GmbH, později zakoupena Sun Microsystems a dnes vyvíjena firmou Oracle. Existují dvě licenční verze tohoto produktu, které jsou dostupné zdarma. První verze je vydaná pro osobní, vzdělávací a komerční použití, která je k dostání pod licencí PUEL. Druhá verze je vydaná pod GPL a je dostupná zdrojovému kódu, tedy open-source.

Aplikace VirtualBox se instaluje přímo na existující operační systém a s tímto řídicím programem může vytvářet další nový hostující operační systémy, které běží ve vlastním virtualizovaném prostředí. VirtualBox podporuje spousty operačních systémů, které se dají vytvářet v tomto prostředí.

**Podporované OS:**

Linux, Mac OS X, Windows XP, Windows Vista, Windows 7 a Solaris, také OSE verze pro FreeBSD.

Co se procesorů týče VirtualBox podporuje práci architektur jak pro Intel s použitím hardwarové virtualizaci VT-x, tak od AMD jeho AMD-V.

VirtualBox, který pracuje přímo s procesorem se snaží, aby hostované stanice pracovaly co nejvíce přirozeně, jako kdyby byly fyzicky nainstalované na hardware. VirtualBox je plnou virtualizací a jádro operačního systému je přesunuto z privilegovaného kruhu 0 do kruhu 1.

Pomocí tohoto řídicího stroje jsme schopni virtuální hosty úplně nezávisle zapínat, pozastavit a zastavovat jednotlivě mezi sebou. Fyzický operační systém a virtualizovaný operační systém mohou mezi sebou komunikovat.

Pevné disky jsou emulovány do diskových obrazů. VirtualBox dovede pracovat s různými formáty, i když byly vytvořené na jiných strojích jiných výrobců, jako například VMWare nebo Microsoft Virtual PC.

**Diskové obrazy je možné používat ve tří různých formátech:**

VDI (Virtual Disk Image) – tento formát je defaultní (specifický, typický formát pro virtualBox)

VMDK – (Virtual Machine Disk Format) – formát kompatibilní s VMware

VHD formát – kompatibilní s Microsoft Virtual PC

Vypnutý virtuální stroj můžeme editovat, tedy měnit jejich nastavený hardware. Například můžeme změnit kapacitu disku, nebo síťový adaptér. Také je možné klonovat disky a importovat nebo exportovat podporované souborové systémy jiných strojů.

Výhodou VirtualBoxu je práce s ISO soubory, které je možné připojit jako virtuální do CD/DVD mechaniky a to se ve virtuálním systému chová jako skutečné CD nebo DVD. Pomocí připojení ISO souboru můžeme snadno nainstalovat operační systém do virtuálního stroje.

Podporovaná grafická karta s běžným nastavením pro VirtualBox je VESA, která je kompatibilní se všemi operačními systémy. Pro náročnější aplikace, který potřebují vyšší výkon, nebo pro dynamické přizpůsobení rozlišení virtualizovaného operačního systému se používá speciální video ovladač.

VirtualBox používá ethernetový síťový adaptér a dokáže virtualizovat několik prostředí pro síťové karty. To dovoluje aby většina operačních systému mohla pracovat bezproblémově bez hledání a instalování nových ovladačů. VirtualBox defaultně nastavuje NAT rozhraní.

VirtualBox dovoluje virtualizovat i zvukovou kartu a to Intel ICH AC'97 nebo SoundBlaster 16.

Podle průzkumu se stal VirtualBox v roce 2007 nejoblíbenějším virtualizačním softwarem pro Linuxové operační systémy. Při testování a porovnávání byly naměřeny lepší výsledky než při použití virtualizační techniky od VMware.(str22)

VirtualBox podporuje USB porty, umožňuje přepínání mezi okny virtuálních systémů a fyzického systému, umožňuje snapshoty. Dále podporuje instalaci 64bitových hostových operačních systémů.

### 3.1.2 VMware Player

VMware player je volně dostupný desktopový softwarový produkt od společnosti VMware a je vyroben pro osobní nekomerční využití. Využívá stejné virtualizační jádro jako produkt VMware Workstation, pro jehož použití je nutné zakoupení licence. Před vydáním verze 3.0 bylo možné vytvořené virtuální obrazy pouze spouštět, ale od verze 3.0 je schopný vytvářet, klonovat a modifikovat. VMware Player se spouští na architektuře x86 a x86-64 a patří do plné virtualizace. Nad touto vrstvou je schopný emulovat dostupné hardwarové zdroje.

VMware Player neumožňuje oproti verzi Workstation importovat nebo exportovat obrazy od jiných produktů.

VMware Player nabízí vyšší výkon pro spolupráci s operačními systémy Windows.

První VMware Workstation byla vyrobena v roce 1999 a v roce 2001 přišla na trh se svou serverovou verzí VMware GSX Server a VMware ESX Server.

#### Výhody VMware Workstation:

- Umožňuje vytvoření snapshotů, které ukládají poslední běžící stav tohoto stroje pod běžícím virtualizovaným operačním systémem, které se dají později obnovit a znovu používat. Virtualizované obrazy je také možné nastavit jako team a propojit je do požadované virtualizované topologie sítě. Stanice takhle mohou mezi sebou komunikovat jako by běžely ve fyzické síti.
- V novější verzi je možné nastavit více procesorů na jednom stroji.
- Instalováním VMware Tools aplikace nám umožní používat drag-and-drop.
- Procesor x64 podporuje obě platformy x86 i x64.

### 3.1.3 KVM a QEMU

#### KVM

Zkratka KVM znamená Kernel-based Virtual Machine. Tento druh nativní virtualizace pracuje pouze na Linuxových operačních systémech a dá se používat na procesorech, které jsou vybaveny technologií Intel VT nebo AMD-V. KVM je licencovány pod GNU licenci a je open-source.

KVM nevykonává žádnou emulaci, protože představuje pouze součást jádra a pro práci s jeho hostovanými virtuálními obrazy využívá rozhraní /dev/kvm. Pro použití KVM je nutná spolupráce s upravenou verzí QEMU.

Původní KVM technologie byla vyvíjena společností Qumranet a v roce 2008 jej koupila společnost Red Hat.

## QEMU

Je plná virtualizace pro architekturu x86. Poskytuje jádro virtualizovanému prostředí a procesorový specifický modul.

Tento emulátor může spouštět několik strojů pro nemodifikované Linuxové nebo Windows obrazy. Každý virtuální stroj má vlastní virtualizovaný hardware – síťovou kartu, disk, grafickou kartu atd.

## 3.2 Příprava testů

Důležitou rolí je výběr testovacích aplikací.

Původním plánem pro testování procesoru bylo naprogramovat vlastní jednoduchý program, který generoval velké pole řetězců délky 1024 bitů a poté přetřídil pomocí třídícího algoritmu quick sort a změřit procesorový čas vykonání. Nicméně program generoval odlišná data v jednotlivých operačních systémech a pro porovnání výkonu je tento program nepoužitelný. Proto je vhodné pro test přidělování CPU procesům použít oficiální testovací benchmark aplikaci. V mém případě vybraný Dhrystone a Whetstone.

Kvalitu výkonu síťové karty, tedy propustnosti jsem měřila aplikací Iperf. Pro práci s diskem, měření diskových operací jako je čtení a zápis malých a velkých souborů jsem zvolila IOZone.

### Hardwarová konfigurace fyzického stroje:

Procesor: Intel Pentium(R) Dual-Core 2.70GHz

Paměť: 2.0 GB

Souborový systém: Linux – ext4, Windows - NTFS

### 3.2.1 Přidělování CPU procesům

#### Whetstone

Whetstone je syntetický benchmark testovací program pro změření výkonu stroje. První verze byla napsaná v roce 1972 v britské (national physical laboratory)

Tento test pracuje s reálnými čísly. Při měření zatěžuje převážně koprocessor a interní L1 cache paměť. Navíc obsahuje transcendentní funkce jako SIN a EXP.

Změřené výsledky jsou zobrazeny v hodnotě milion Whetstone instrukcí za sekundu (MWIPS – Million of Whetstone Instruction Per Seconds).

## **Dhrystone**

Původně vyvinutý v roce 1984 R.P. Weckerem je open-source program pro testování výkonu procesoru. Aplikace je napsaná v programovacím jazyku C nebo Pascal a v dnešní době dostupná i v jazyku Java a testuje výkon s celočíselnými výpočty. Ve svém měření jsem použila 32-bitovou verzi (2.1).

Program je procesorově vázaný, testuje pouze výkon procesoru a ne paměti, protože celý test se vejde do poměrně malé cache (asi 8 kB).

Výsledky jsou vyhodnoceny v Dhrystonech MIPS, to znamená, kolik mikro instrukcí proběhne za sekundu.

### **3.2.2 Diskové operace**

#### **IOZone**

IOZone je nástroj pro měření výkonu vstupních/výstupních souborů souborového systému. Pracuje způsobem kdy generuje a měří několik různých druhů I/O souborových operací. IOZone dovede pracovat na spoustě strojích a operačních systémech. Práce je zahrnuta v prostředí příkazového řádku. Při testování je možno použít velké množství druhů parametrů a tím striktně nastavit jakým způsobem bude test probíhat. Zahrnuje i možnost kdy se automaticky vygeneruje Excelový soubor s naměřenými výsledky, které jsou zobrazeny v grafu.

#### **Můžeme provádět následující operace:**

Read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read/write, read/pwrite variants, aio\_read, aio\_write, mmap.

#### **Obecně se dají použít tři módy:**

1. Regulární
2. Propustní
3. Cluster

Regulární mód je výchozí a využívá k testování jedno vlákno. Propustný mode dovoluje uživateli specifikovat kolik má k testování použít vláken. Je možné, že někdy z nějakého důvodu regulární mód navrátí odlišné výsledky, než při použití propustnostního módu s jedním vláknem. Cluster mód používá speciální konfigurační soubor, který se dá využívat pro více experimentů.

Při měření výkonu se dají nastavit různé velikosti testovacích souborů. Výsledkem testování jsou spousty naměřených výstupních údajů.

### **Příklad použití:**

Nejsnadnější způsob pro měření je automatický test.

```
Iozone -a
```

Tento test je schopný vygenerovat graf

```
Iozone -Ra
```

Tento test je omezený, nevykonává všechny testy, pouze provede čtení a zápis. Rychlé měření výkonu.

```
Iozone -Ra -g 2G -i 0 -i 1
```

### **Definice testů:**

**Write:** Tento test měří výkon při zápisu nového souboru. Také ukládá informaci o umístění nově vytvořených dat na disku. Obsahuje informace o adresáři, alokaci místa a další data spojená se souborem, který není součástí dat obsažených v souboru. Tyto informace se nazývají metadata.

**Re-write:** Měří výkon při zápisu souboru, který už byl jednou zapsán. Výsledkem re-write testů bývá vyšší výkon než u zápisu nového souboru.

**Read:** Testuje výkon při čtení existujícího souboru.

**Re-Read:** Testuje výkon při čtení souboru, který už byl testovaný.

**Random Read:** Test měří výkon při čtení souborů, který jsou náhodně umístěny v souboru.

**Random Write:** Test měří výkon při zápisu souborů, které jsou náhodně umístěny v souboru.

**Random Mix:** Test měří výkon při čtení a zápisu souborů, které jsou náhodně umístěny v souboru. Tento test je dostupný pouze při testování propustnosti.

**Backwards Read:** Test měří výkon při čtení souboru pozpátku.

**Record Rewrite:** Test měří výkon při zápisu a opakovaném zápisu.

**Fwrite:** Při měření výkonu zápisu souboru se používá knihovna s funkcí fwrite().

Je běžně dostupná v knihovně, kde vykonává zápis operací do vyrovnávací paměti. Tento test se zapisuje na již existující soubor.

**Fread:** Tento test měří výkon při čtení souboru s použitím fread() funkce z knihovny

**Freread:** Test měří stejné vlastnosti jako u fread testu a navíc zahrnuje testování souboru, který už byl čtený.

### **Speciální testy:**

**Mmap:** Většina operačních systémů podporuje použití funkce `mmap()` aby namapovala soubor do uživatelského adresního prostoru. To se používá pokud chceme, aby se pracovalo se soubory jako s kusem paměti.

**Async I/O:** Další způsob měření, který se dá používat pod mnoha operačními systémy je POSIX async I/O. Tato aplikace používá POSIX standard a async I/O rozhraní.

### **Pravidla používání:**

- Abychom získali přesné výsledky je potřeba zajistit, aby maximální velikost souboru byla větší než velikost vyrovnávací paměti. Také je možné použít maximální velikost souboru, aby bylo dosaženo větší velikosti než fyzické paměti. To je možno použít i v případě dynamické vyrovnávací paměti.
- Velikost souboru se přizpůsobuje velikosti vyrovnávací paměti procesoru.
- Velikost souboru se přizpůsobuje vyrovnávací paměti.
- Velikost souboru je větší než velikost vyrovnávací paměti.

### **Výhody:**

- Dostupnost zdrojového kódu.
- Freeware verze - používání zdarma.

### **Některé parametry a jejich vlastnosti:**

- a      automatický mód
- b file name      vytvoření binárního souboru
- g #      nastavení maximální velikosti souboru pro automatický mód
- i #      specifikace testování. (0 = write/rewrite, 1=read/re-read, 2=random-read/write, 3=Read-backwards, 4=Re-write-record, 5=stride-read, 6=fwrite/re-fwrite, 7=fread/Re-fread, 8=random mix, 9=pwrite/Re-pwrite, 10=pread/Re-pread, 11=pwritev/Re-pwritev)
- l #      nastavení nejnižšího limitu pro počet procesů, které se mají vykonat
- n #      nastavení minimální velikosti souboru pro automatický mód
- N      výstup výsledků v mikrosekundách za operaci
- R      generuje excelový výstup
- v      zobrazí verzi programu
- u #      nastavení nejvyššího limitu pro počet procesů, které se mají vykonat



### 3.2.3 Propustnost sítě

#### Iperf

Je síťově používaný testovací nástroj pro měření propustnosti a kvality síťového spojení. Iperf je psaný v programovacím jazyce C++. Existuje i verze pro GUI, která je napsaná v jazyce Java.

Patří mezi open-source software a můžeme ho používat na několika různých platformách a to UNIX/Linux nebo Microsoft Windows.

Při měření vytváří TCP a UDP datové toky. Kvalitu spojení měříme podle zpoždění, neboli času odezvy, podobně jako u příkazu PING. Abychom si ujasnili rozdíl mezi TCP a UDP protokoly, TCP používá algoritmy, které slouží ke kontrole paketů a jejich správnému doručení příjemci, zatímco UDP posílá pakety do sítě aniž by kontroloval jejich správné doručení. Říkáme, že TCP je protokol spojitý a spolehlivý, kdežto UDP patří mezi nespojitý a nespolehlivé protokoly. Výhodou UDP protokolu je rychlejší přenos než pomocí TCP.

Měření probíhá způsobem, kdy je síť rozdělena na dvě stanice – serverovou a klientskou. Iperf aplikace se tedy spouští na jedné straně jako server a na druhé straně se pomocí parametru a IP adresy serveru připojí jako klient. Poté ihned začne testování. Po změření vidíme výsledky na obou stranách.

Výstup obsahuje oznámení o množství přenesených dat a změřené propustnosti za určitý časový interval.

Výhodou této aplikace je, že ji můžeme použít kříženě na různých platformách, proto se nejčastěji používá pro porovnávání drátových a bezdrátových síťových zařízení.

## 3.3 Měření a vyhodnocení

Výsledné hodnoty z měření jsou uloženy v souboru, který je přiložen k této bakalářské práci. Měření probíhalo způsobem, že jsem testy provedla vždy nejprve na fyzickém stroji a poté postupně na produktech pro desktopovou virtualizaci.

### 3.3.1 Měření přidělování CPU procesům

Po změření výkonu procesoru pomocí aplikací Whetstone a Dhrystone jsem zpracovala výsledky a uvedla je v grafech, viz. níže. Vzhledem k závislosti na operačním systému a použití kompilátoru pro překlad kódu, jsem rozdělila toto měření do čtyř grafů.

#### Whetstone

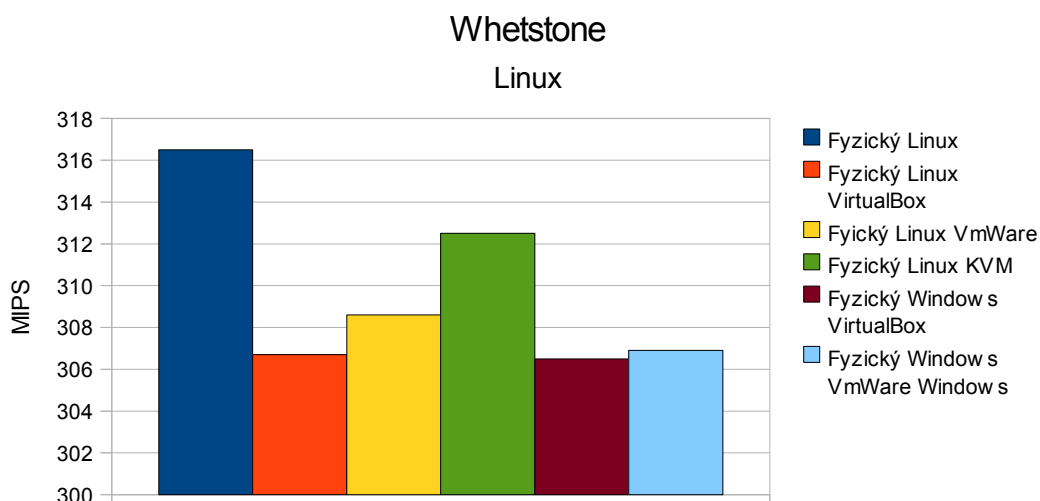
Výsledky prvních dvou grafů jsou rozděleny podle testovaného operačního systému (Linux nebo Windows) a jsou uvedeny v hodnotách MWIPS (Million of Whetstone instruction per second).

## Měření a hodnocení

V grafu č. 1 můžeme vidět, že rychlost virtuálního stroje, kdy fyzickým operačním systémem je Linux, vyšlo nepatrné zpomalení. Virtuální stroj tedy může v průměru maximálně využít 97% z procesoru.

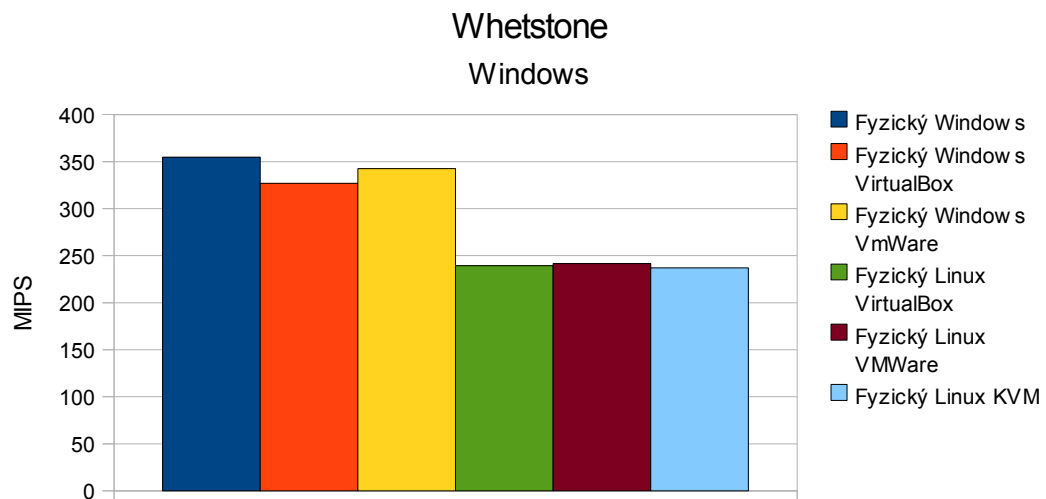
Virtuální stroj s nejlepšími výsledky pro testování procesů je KVM a VmWare.

Nastavení VMware nám dovoluje využít funkcionality nových Intel/AMD procesorů pro používání virtualizace. Použijeme-li tuto techniku pro virtualizování procesoru, můžeme tak zvýšit výkon nativního stroje až o 36%



Graf 1: Whetstone Linux

Z grafu 2 opět vidíme, že používáme-li tentýž fyzický a virtuální stroj stejného fyzického operačního systému, výsledkem je nepatrné zpomalení. Naopak, pokud měřený virtuální stroj běží na linuxovém operačním systému vidíme zpomalení o 30%.



Graf 2: Whetstone Windows

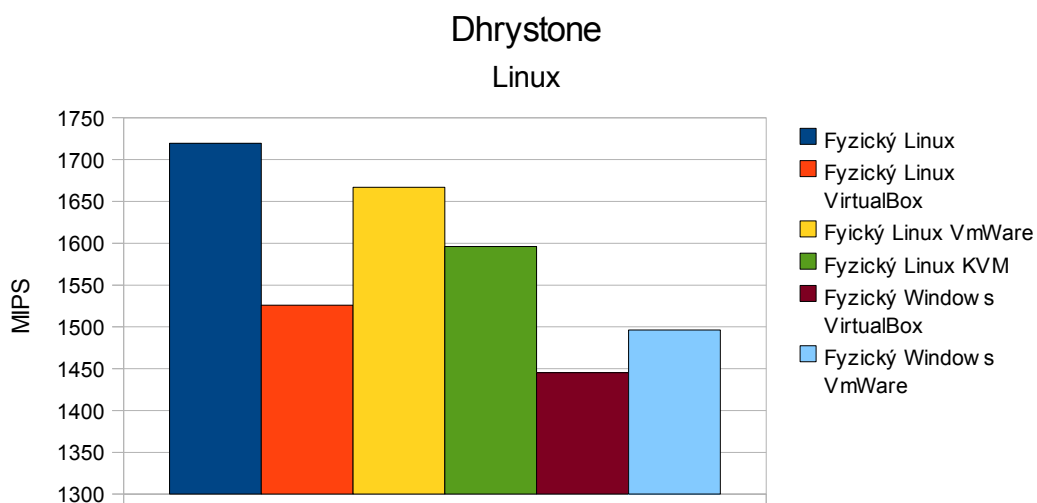
### Dhrystone

Další dva grafy zobrazují naměřené hodnoty pro operační systémy Linux a Windows při používání testovací aplikace Dhrystone. Výsledky jsou zobrazeny v Dhrystonech za sekundu.

### Měření a hodnocení

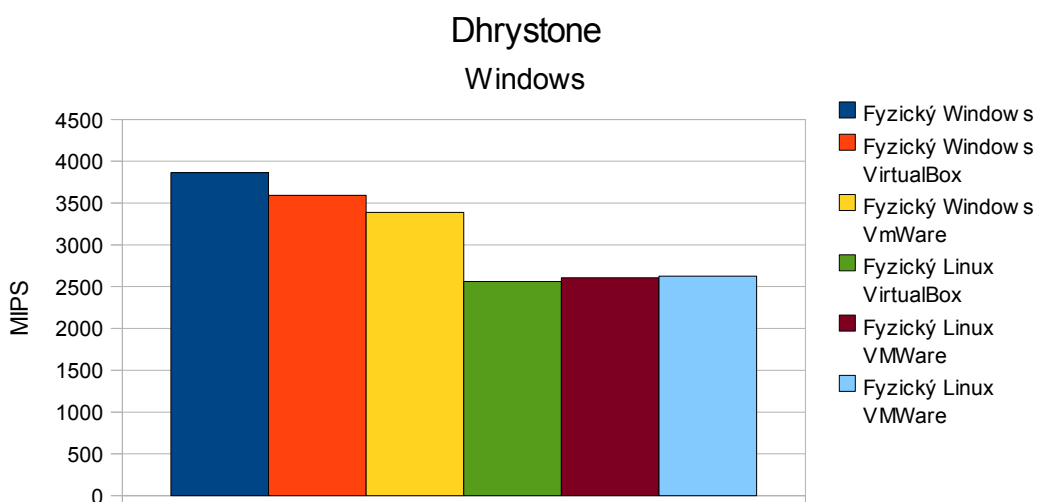
Měření procesorového výkonu pomocí Dhrystone viz. graf 3 na linuxovém operačním systému vychází obdobně jako v případě měření Whetstone z grafu 1.

Virtuální stroj s nejlepšími výsledky pro testování procesů operačního systému je opět KVM a VmWare.



Graf 3: Dhrystone Linux

Poslední graf č. 4 zobrazuje výsledky testované na strojích s operačním systémem Windows pomocí Dhrystone benchmark aplikace. Výsledky jsou opět zobrazeny v Dhrystonech za sekundu. Opět vidíme, že zpomalení fyzického a virtualizovaného systému Windows je zhruba 97% z plného výkonu procesoru.



Graf 4: Dhrystone Windows

### **Zhodnocení pro CPU procesy:**

Jednotlivé produkty virtuálních strojů pro desktopové operační systémy neovlivňují výkon pro práci s procesorem. Při používání stejného operačního systému pro fyzický i virtualizovaný stroj podává virtualizovaný systém lepší výkony, než při používání systémů kříženě.

### **3.3.2 Propustnost sítě měřená pomocí Iperf aplikace**

Pro měření výkonu propustnosti sítě jsem použila síťovou kartu s Gigabitovým Ethernetem (přenosová rychlost 1Gbit/s) a testovala jsem vždy jeden virtuální stroj stejného operačního systému jako fyzický stroj. Obě stanice byly přímo zapojeny do P2P sítě kříženým UTP kabelem kategorie 5e.

Testovací aplikace Iperf s využitím transportního řídicího protokolu TCP vytvořila spojení pro přenos, kde na straně fyzického stroje byla spuštěna jako server a na straně virtuálního stroje jako klient.

U virtuálního stroje jsem měřila dva druhy testů. V prvním testu se měřil virtuální stroj, který byl připojený do sítě jako NAT. Což je jednodušší způsob připojení stroje do sítě, protože nevyžaduje žádnou konfiguraci fyzického ani virtuálního stroje. Toto připojení funguje jako router. Virtuální stroj je implicitně nastavený, aby získal potřebnou konfiguraci z fyzického stroje pomocí DHCP, který se používá pro automatické přidělení IP adres. V našem případě je virtuální stroj připojený transparentně a tím vstupuje a vystupuje do sítě. Dalo by se říct, že virtuální stroj se za routerem jeví jako v privátní síti, což v některých případech může být nevýhodou, protože není viděn v síti.

Řešením takového problému je právě druhý způsob měření a to pomocí metody BRIDGE (česky most).

S mostovým síťováním využívá fyzický stroj zařízení ovladače síťového adaptéru. To dovoluje virtuálnímu stroji, aby oddělil data z fyzického stroje a využil je uvnitř. Efektivně se tak vytvoří nové síťové rozhraní v softwaru. Pokud virtuální stroj používá nové softwarové rozhraní, jeví se fyzickému systému jakoby by byl host fyzicky připojený do rozhraní pomocí síťového kabelu. Fyzický systém přes toto rozhraní posílá a přijímá data hostu. To znamená, že můžeme nastavit routování nebo mostění mezi virtuálními stanicemi a zbytkem naší sítě a poté s nimi komunikovat. Každé zařízení je potom jedinečné v síti a musí mít svojí IP adresu a název počítače.

VirtualBox obsahuje několik prostředí pro nastavení síťové karty a pro každou kartu můžeme v pokročilém nastavení inividuálně nastavit jaký typ síťové karty budeme používat ve virtuálním stroji.

**Prostředí sítě, která se dají simulovat ve virtualBoxu:**

- Virtualbox
- Nepřipojeno
- NAT
- Síťový most
- Vnitřní síť
- Pouze s hostem

**VirtuaBox může virtualizovat tyto typy síťových karet:**

- AMD PCNet PCI II (Am79C970A);
- AMD PCNet FAST III (Am79C973, the default);
- Intel PRO/1000 MT Desktop (82540OEM);
- Intel PRO/1000 T Server (82543GC);
- Intel PRO/1000 MT Server (82545EM);
- Paravirtualized network adapter (virtio-net).

V prvních testech jsem proměřila výkon všech typů síťových karet. Pro měření jsem zvolila typ Intel PRO/1000 MT Desktop, protože výsledky ostatních testů byly pomalejší, pro naše měření zanedbatelné.

VMware player využívá nastavení podobné, ale neobsahuje možnost výběru z několika typů síťových karet, využívá se výchozího nastavení.

Výchozí nastavení používání KVM pro nastavení sítě s příkazem je síťová karta `ne2k_pci` a NAT prostředí.

```
-net nic -net user
```

Pomocí parametrů můžeme nastavit tyto druhy:

- `pcnet`
- `rtl8139`
- `virtio`

**Prostředí pro síťovou virtualizaci:**

Bridged – přímé připojení do sítě

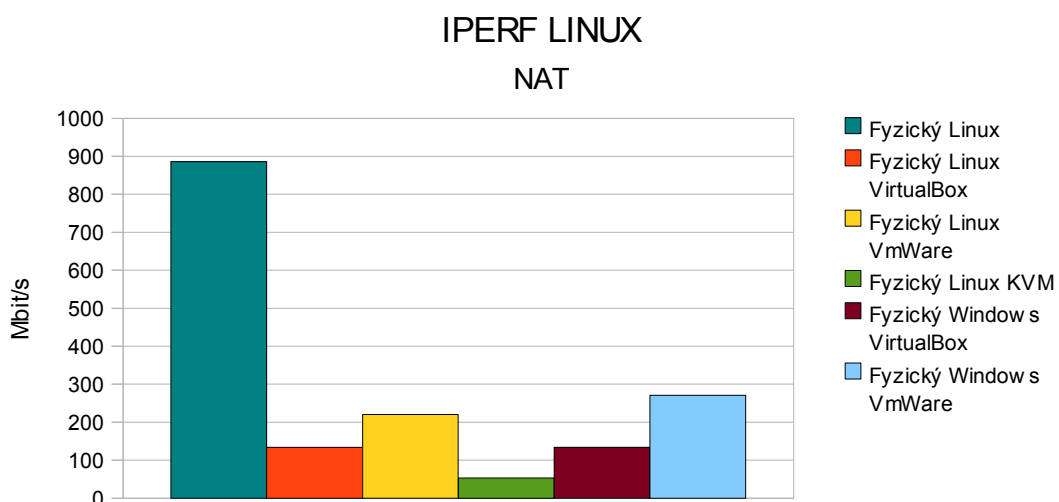
NAT – využívá sdílenou IP adresu fyzického stroje

Host only – veřejná síť, sdílení pouze s fyzickým strojem

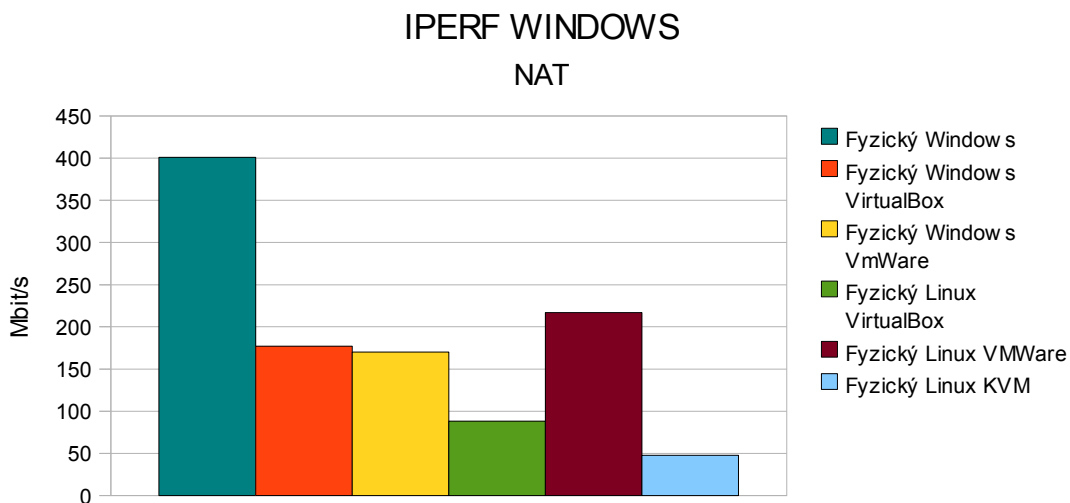
Pro používání KVM je jednodušší metodou používání NAT prostředí. Chceme-li využívat prostředí bridge, musíme nastavit fyzické nastavení sítě, které se přizpůsobí tomuto použití.

Výsledné hodnoty můžeme porovnat ve čtyřech grafech, které jsou zobrazeny v Mbit/sec. První dva grafy reprezentují síťové připojení NAT pod Linuxem graf č. 5 a Windows graf č. 6.

#### Měření výkonu sítě s NAT připojením:



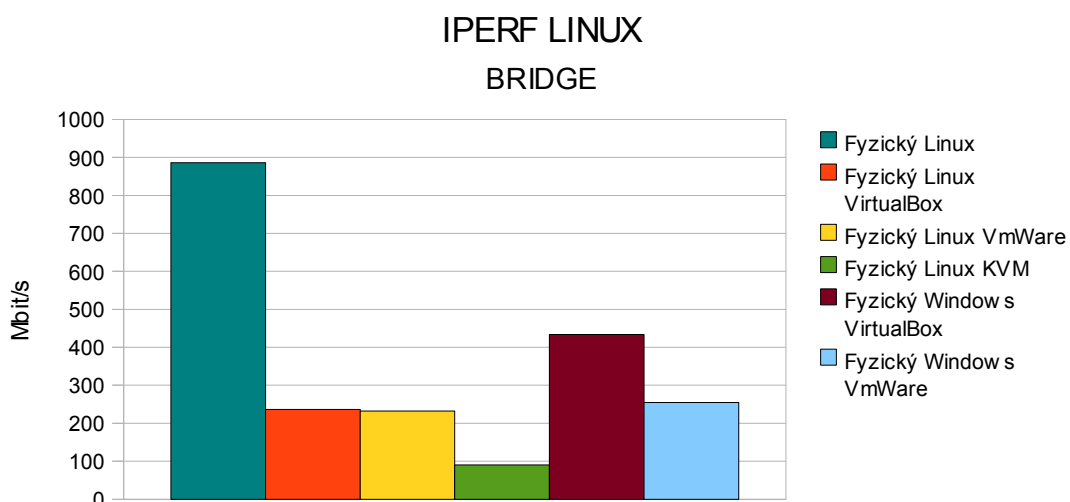
Graf 5: Iperf Linux NAT



Graf 6: Iperf Windows NAT

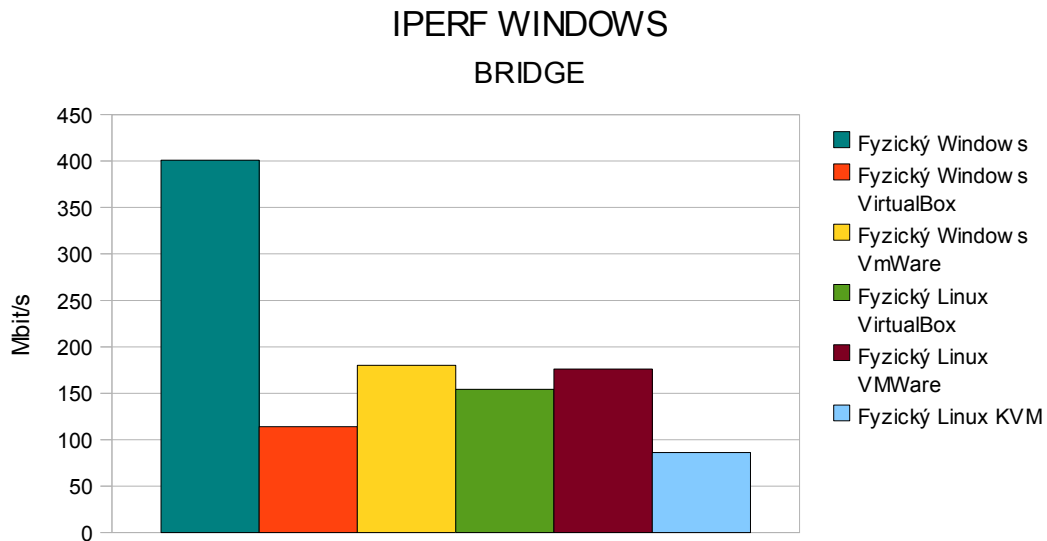
Další dva grafy reprezentují připojení virtuálního stroje do sítě pomocí bridge rozhraní. Graf 7 zobrazuje výsledné hodnoty pro Linuxové stroje a graf 8 pro stroje s operačním systémem Windows.

#### Měření výkonu sítě s mostovým připojením:



Graf 7: Iperf Linux Bridge





Graf 8: Iperf Windows Bridge

#### Zhodnocení:

Ze všech výsledných grafů vidíme značný pokles výkonu síťové propustnosti virtuálních strojů.

Přestože naměřené hodnoty nedosahují nad 50% využití výkonu síťové karty, je toto použití rychlosti dostačující. Zhodnotíme-li výkon jednotlivých virtuálních strojů v grafech vidíme, že výkon síťové karty je lepší pro Bridge rozhraní než pro NAT. KVM stroje vychází vždy s nejhoršími naměřenými výsledky.

Propustnost ve Windows je oproti Linuxu nižší, protože Windows 7 obsahuje nové, vyšší zabezpečení sítě.

### 3.3.3 Diskové operace měřené pomocí IOZone

Posledním měřením výkonu virtuálních strojů je testování diskových operacích pomocí IOZone.

#### Příklad použitého příkazu:

```
iozone -ec -t 1 -r 4k -s 128m -+n -i 0 -i 1 -R
```

#### Použité parametry a jejich vlastnosti:

- r specifikuje velikost záznamu k testování 4 Kbytes
- s specifikuje velikost souboru pro testování 128 MB
- i 0 specifikuje write/rewrite pro testování
- i 1 specifikuje read/re-read
- t běží v propustném módu, dovoluje specifikovat kolik vláken nebo procesů má aktivovat během měření

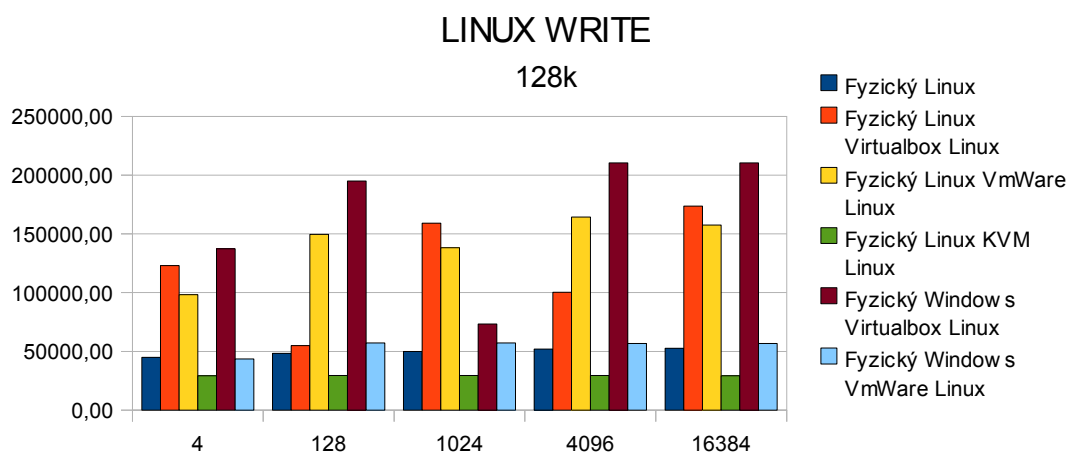
Testování propustnosti s jedním vláknem pro čtení a zápis malého (128 MB) a velkého (4 GB) souboru v aktuálním adresáři. V testování je zahrnuto fsync a nevykonává opakované testování.

Velikost záznamu se měnila 4/64/512/2048/16384 kBytes.

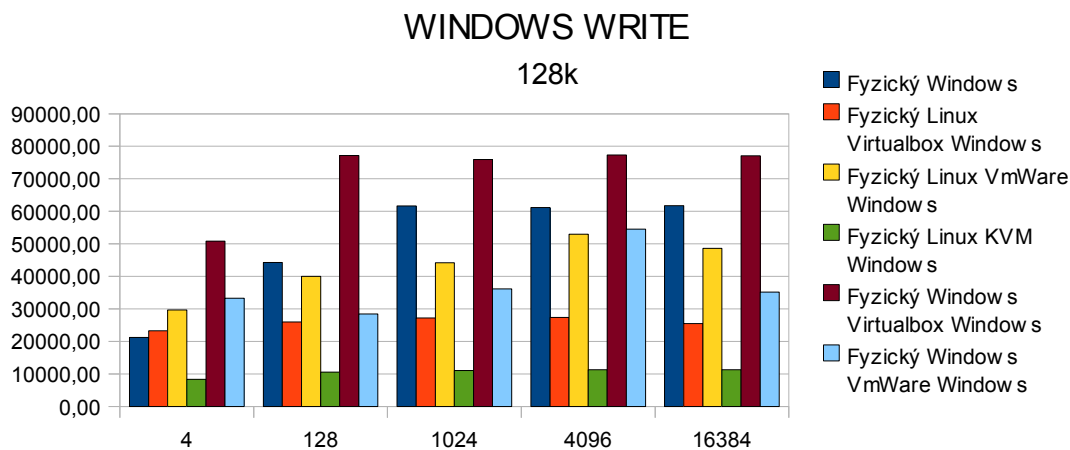
### Výsledky:

Naměřené hodnoty jsou zpracované a zobrazené v grafech podle velikosti testovaného souboru tedy 128 MB nebo 4 GB a podle operačního systému. Vzhledem k tomu, že se testovalo čtení i zápis, ve výsledku vzniklo 8 grafů. Naměřené hodnoty jsou zobrazeny v Kbytes/sec.

První dva grafy 9 a 10 jsou vyhodnoceným testem souborového systému pro práci se 128 KB souborem. Vidíme, že některé virtuální stroje dosahují vyššího výkonu pro zápis malého souboru. Stroj s nejhoršími výsledky je v obou případech KVM a stroj s nejlepšími fyzický Windows VirtualBox Windows.



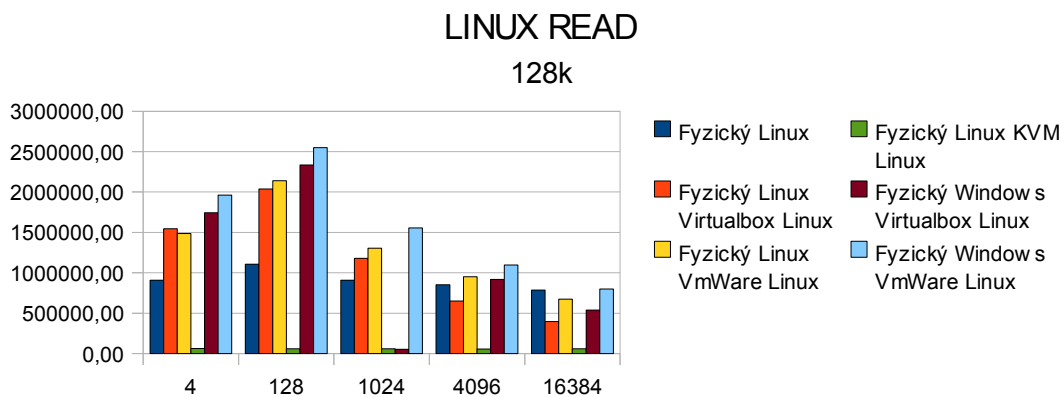
Graf 9: Linux Write 128km



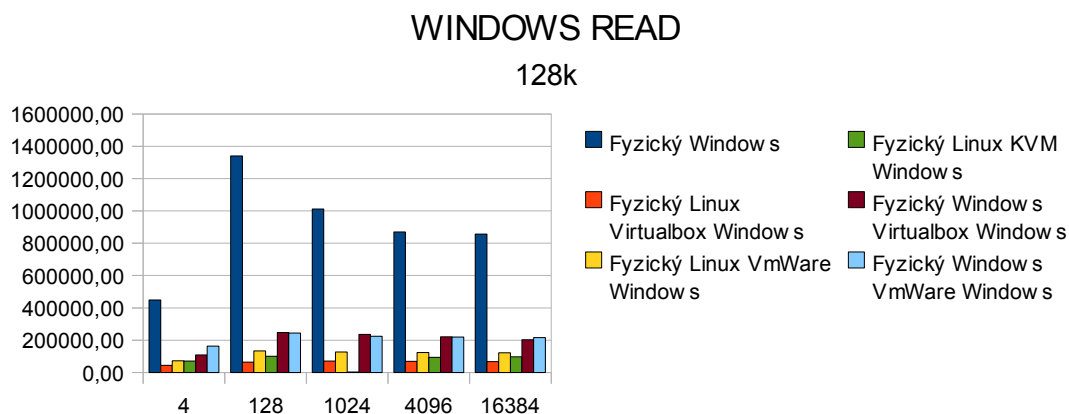
Graf 10: Windows Write 128k

Další dva grafy č.11 a č.12 zobrazují čtení výkonu souborového systému pro Linuxový a Windows operační systém.

Z grafů vidíme opět jako nejhorší stroj pro čtení malého souboru virtuální stroj KVM. Je zajímavé, že Linuxové stroje dosahují mnohem lepšího výkonu než Windows.



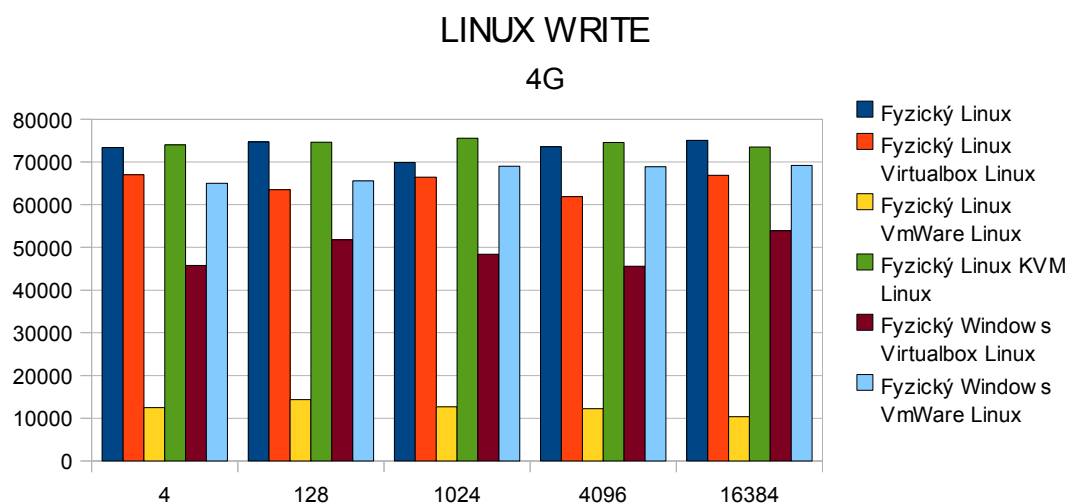
Graf 11: Linux Read 128k



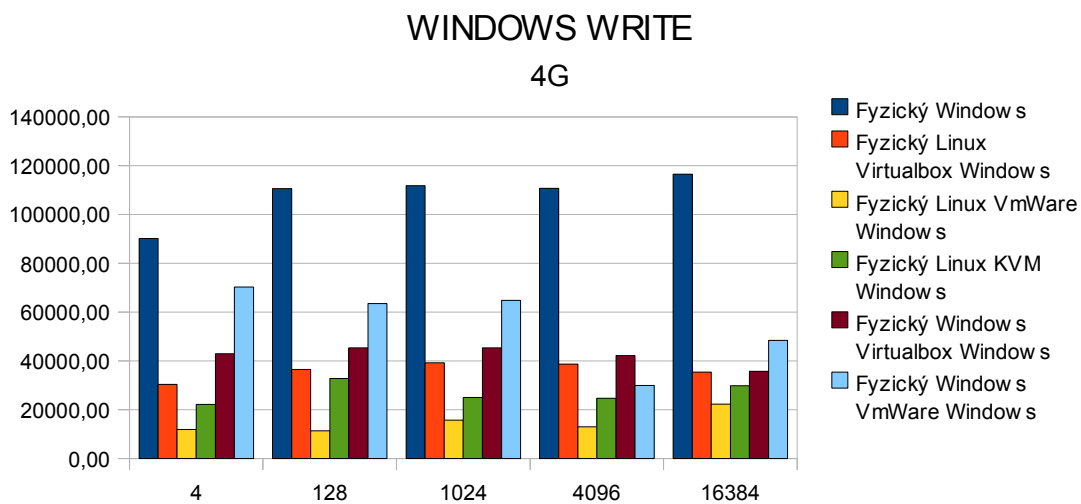
Graf 12: Windows Read 128k

Grafy č. 13 a 14 jsou výsledkem práce pro zápis velkého 4 GB souboru, rozdělené podle operačních systémů.

Podle naměřených výsledků je nejhorším strojem s fyzickým operačním systémem Linux a virtuálním VMware Linux a fyzický systém Linux a VMware Windows. U zápisu 4GB souboru značně vede fyzický operační systém Windows.

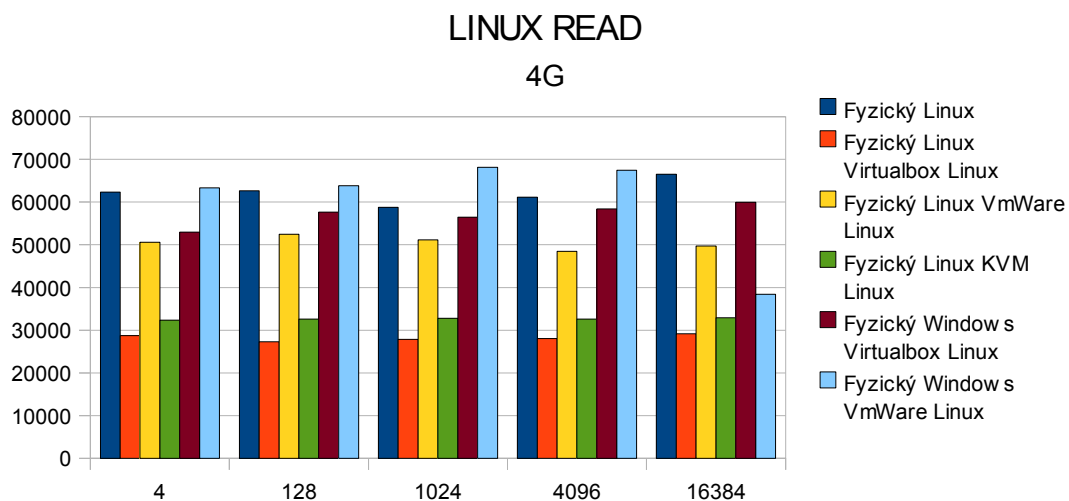


Graf 13: Linux Write 4G

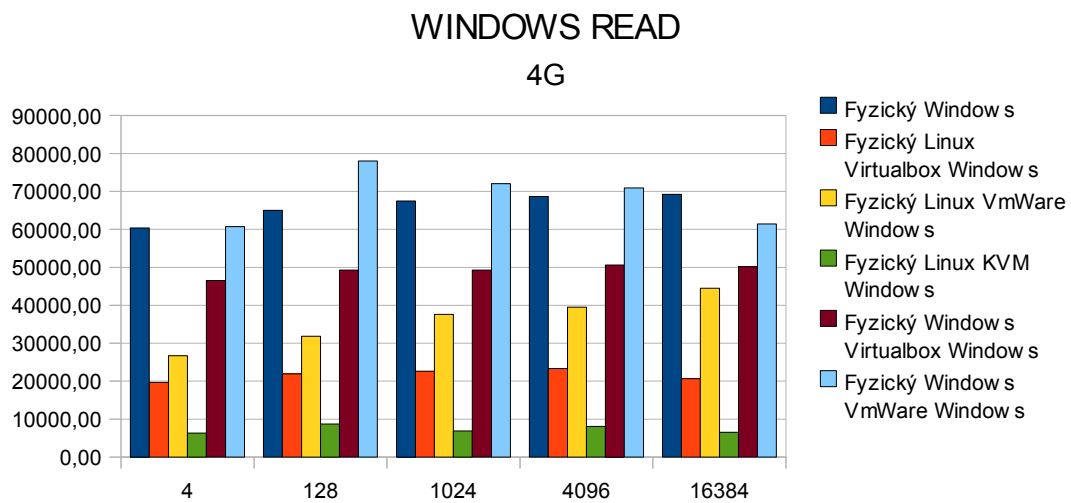


Graf 14: Windows 4G

V posledních grafech č. 15 a 16 vidíme naměřené výkony dvou operačních systémů pro čtení velkého 4 GB souboru.



Graf 15: Linux Read 4G



Graf 16: Windows Read 4G

## 4. Závěr

Očekávaným výsledkem této práce bylo po provedení tří vybraných testů na všech virtuálních strojích zpomalení virtuálních strojů oproti fyzickému. Po poměření výkonu těchto strojů se tak naše očekávání splnilo.

Zjistili jsme, že při testování CPU procesů je výkon nepatrně nižší, než při fyzickém stroji. Nicméně propustnost sítě, testovaná na gigabitovém Ethernetu vyšla s neočekávaně velkým zpomalením. Výsledky diskových operací vyšly ve většině případů zpomalením pro Windows operační systémy.

Jako pokračování této práce by bylo velmi zajímavé navázat s těmito testy i na měření výkonu pro serverové operační systémy.

## 5. Použité zdroje

- [1] Techniky virtualizace, Internet  
<http://cs.wikipedia.org/wiki/Virtualizace>
- [2] Techniky virtualizace, Internet  
<http://en.wikipedia.org/wiki/Virtualization>
- [3] Plná virtualizace, Internet  
[http://en.wikipedia.org/wiki/Full\\_virtualization](http://en.wikipedia.org/wiki/Full_virtualization)
- [4] Částečná virtualizace, Internet  
[http://en.wikipedia.org/wiki/Partial\\_virtualization#Partial\\_virtualization](http://en.wikipedia.org/wiki/Partial_virtualization#Partial_virtualization)
- [5] Paravirtualizace, Internet  
<http://en.wikipedia.org/wiki/Paravirtualization>
- [6] Harwarově asistovaná virtualizace, Internet  
[http://en.wikipedia.org/wiki/Hardware-assisted\\_virtualization](http://en.wikipedia.org/wiki/Hardware-assisted_virtualization)
- [7] OVF, Internet  
[http://en.wikipedia.org/wiki/Open\\_Virtualization\\_Format](http://en.wikipedia.org/wiki/Open_Virtualization_Format)
- [8] Dhrystone, Internet  
<http://www.webopedia.com/TERM/D/Dhrystone.html>
- [9] KVM, Internet  
<http://www.linux-kvm.org/>
- [10] Virtualbox, Internet  
<http://www.virtualbox.com>
- [11] VMware, Internet  
<http://www.vmware.com>
- [12] IOZone, Internet  
<http://www.iozone.org/>
- [13] Iozone dokumentace a ukázkový příklad, Internet  
[http://www.iozone.org/docs/IOzone\\_msword\\_98.pdf](http://www.iozone.org/docs/IOzone_msword_98.pdf)
- [14] Dhrystone a Whetstone, Internet  
<http://www.roylongbottom.org.uk/index.htm#anchorCPU>
- [15] Iperf, Internet  
<http://www.filewatcher.com/m/iperf.exe.110592.0.0.html>